

7-13-00

75

PATENT

Express Mail No. EL389673485US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTOR: Karen R. Kluttz
Sandeep K. Singhal
Thyra Rauch
David A. Schell

APPLICATION NO. Not Yet Assigned

FILED: Herewith

CASE: RSW-00-0021

TITLE: METHOD AND APPARATUS FOR LEARNING
COMPUTER INTERFACE ATTRIBUTESCERTIFICATE OF EXPRESS MAILING

I hereby certify that this correspondence, along with any papers indicated as being enclosed, is being deposited as Express Mail (Label No. EL389673485US) in an envelope addressed to: Box Patent Application, The Commissioner for Patents, Washington, D.C. 20231, on July 12, 2000.

Date 7/12/00Douglas M. Cook
Douglas M. Cook

BOX PATENT APPLICATION
Commissioner for Patents
Washington, DC 20231

NEW APPLICATION TRANSMITTAL LETTER

Sir:

Enclosed are the following papers relating to the above-named new application for patent:

1. Specification (22 pgs., including claims and abstract);
2. Informal Drawings (4 sheets);
3. Declaration and Power of Attorney (3 pgs.);
4. Assignment (2 pgs.);
5. Assignment Cover Sheet (4 pgs.)



09/06/00 09:42:00

CLAIMS AS FILED				
	No. Filed	No. Extra	Rate	Calculations
Total Claims	25 - 20 =	5	\$18	\$90.00
Independent Claims	2 - 3 =	0	\$78	\$0.00
Multiple Dependent Claim(s), if applicable			\$260 =	\$0.00
Basic Filing Fee				\$690.00
Total Fee:				\$780.00

Please file the application and charge **International Business Machines (IBM)** Deposit Account No. **09-0461** the amount of **\$780.00** to cover the filing fee. Two copies of this letter are enclosed. In the event of non-payment or improper payment of a required fee, the Commissioner is authorized to charge or to credit **Deposit Account No. 09-0461** as required to correct the error.


Please address all correspondence to:

**Theodore Naccarella
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107-2950**

Telephone calls should be directed to the undersigned at (215) 923-4466.

Respectfully submitted,

7.12.00
Date


Theodore Naccarella
Registration No. 33,023

Attorney for Applicant
International Business Machines
Corporation

PATENT

1

Docket No. RSW-00-0021

METHOD AND APPARATUS FOR LEARNING
COMPUTER INTERFACE ATTRIBUTES

Field of the Invention

5 The invention pertains to operator interfaces for
computer displays. More particularly, the invention pertains
to a method and apparatus for automatically storing and
recalling computer file display preferences.

Background of the Invention

10 Frequently, workers in offices spend significant amounts
of time using computers. Many of the tasks performed by such
workers are repetitive. As merely an example, let us consider
a data entry clerk for a financial institution, such as a
bank. One aspect of this individual's responsibilities
includes transferring information from one computer file to
15 another. For example, let us consider a bank in which
information about its customers had previously been stored in
signature cards which have now been scanned and stored as

image files in a computer database. Such information might include the customer's name, address, social security number, mother's maiden name, and a copy of the customer's signature.

As part of a scheme to automate customer banking, the bank has scanned all of its signature cards into a computer and converted them into .gif format (Graphic Interface File) files. Each signature card has been converted into two image files representing the front and the back of each card, respectively. The front image file is named #name.gif and the back image file is named #name.back.gif, wherein "#name" represents the customer's actual name. It is the data entry clerk's responsibility to open up each of these .gif files and manually type in data from those signature cards into a database using a database application program. Accordingly, for each customer, the data entry clerk (1) opens the .gif file corresponding to the front of the signature card, (2) opens the file corresponding to the back of the same signature card, (3) sizes the windows displaying the two files so as to take up the top left hand quadrant of the screen and the bottom left hand quadrant of the screen, respectively, (4) opens up another window which provides a GUI (Graphical User Interface) in which the data entry clerk can type in information from the two aforementioned .gif files, (5) sizes and positions that window to take up the right-hand half of the screen, and (6) enters the data shown in the signature card files displayed in the left hand half of his computer

screen into the GUI which is on the right hand half of the screen. These steps must be repeated for every customer.

It is well known in the prior art to provide a operator interface program that automatically sizes application program windows based on operator preferences. For instance, most software packages, such as those that run under the Microsoft™ Windows operating system, have the ability to size and position themselves in windows that are the same size and position that the programs had been when they were last closed. For example, if the last time a computer operator opened a particular word processing application program on a particular computer, the user interface window for that program was positioned and sized in a particular manner, the next time the program is opened, it opens up in a window of the same size and position. This particular feature is also used with respect to other windows within that program. Thus, for instance, if the last time a user closed a sub-window, such as a sub-window for selecting a file to be opened, that window was sized to one quarter the total screen size and positioned in the upper right hand corner, the next time the user selects to open a pre-existing file, the corresponding sub-window appears with the same size and position. If the user resizes or repositions the sub-window before closing it, the operating system software remembers the latest size and position. The next time that particular sub-window is opened, it will be opened with the latest size and position. However,

documents (i.e., computer files) that are opened within the application program are opened to a standard size each time.

Accordingly, it is an object of the present invention to provide an improved operator interface for a computer display.

5 It is a further object of the present invention to provide a method and apparatus for automatically assigning desirable display attributes to files opened with an application program.

10 It is yet another object of the present invention to provide a method and apparatus for automatically opening files of particular types having desirable display attributes particular to that type.

15 It is one further object of the present invention to provide a method and apparatus for automatically opening files of particular types having desirable display attributes as dictated by the operator's preferences.

Summary of the Invention

20 In accordance with the present invention, a feature is provided in which the application program stores information about certain operator display preferences for particular file types used in that application and then automatically applies those preferences whenever files of the same type are opened. Thus, in accordance with the invention, display attributes that can be set by the operator, such as window size, window
25 position, font, and color, are stored in a table associated

with the particular file type. Whenever a file of the same type is opened, it is assigned the same values for those attributes. The file type may be given by one or more extensions in the file name, such as .gif, .tif, .doc, .wpd.

5 The invention can be expanded to include automatically opening a second file of a related type whenever a file of the first type is opened and sizing, positioning and assigning it even further attributes as dictated by the operator's preference.

10 In accordance with the invention, the operator may be given the option to manually set the attributes for a given file type. Alternately, the existing attribute values at the time when a file is closed may be automatically assigned to the next file of the name type that is opened.

15 **Brief Description of the Drawings**

Figure 1 is a pictorial view of an exemplary computer screen display having a first set of display attributes.

Figure 2 is a pictorial view of an exemplary computer screen display having a second set of display attributes.

20 Figure 3 is a flow chart illustrating operation in accordance with one particular embodiment of the present invention.

Figure 4 is a flow chart illustrating operation in accordance with another particular embodiment of the present invention.

25

Detailed Description of the Invention

Figure 1 is a copy of an exemplary computer display screen 10 in which an operator of the computer has opened an application program and several documents 12, 14, 16, 18 within that program in accordance with the prior art. Particularly, the program has opened into a window which consumes the entire display screen. Within the program, the operator has opened up an image file 12 named johnsmith.gif. It is an image of the front of a signature card of an individual named, John Smith. The operator also has opened up a second file 14 which is the back of the signature card. This file is named johnsmith.back.gif.

In accordance with the protocol of the bank for which the operator works, signature card images are assigned file names as follows: the front of each signature card is named #customername.gif, and the back of each signature card is named #customername.back.gif, where "customername" is the actual name of the customer. Accordingly, it is known that the file corresponding to the back of the signature card whose front is stored as johnsmith.gif is named johnsmith.back.gif.

The operator also has opened up third and fourth files 16, 18 in which he or she will enter data. In Figure 1, the operator has merely opened the files in accordance with the application program, but has not yet resized or repositioned the windows or changed any other attributes of the display of the files. Accordingly, Figure 1 shows the computer screen

condition after opening these four files as set by the default conditions of the application program. The operator then manipulates each of the four windows to size and position them in a convenient manner for the task that he or she is to perform. Thus, for instance, the user may manipulate the four windows to the positions and sizes shown in Figure 2. The manner of manipulating windows to resize and reposition them is well known to those in the art and involves much manipulation with one's mouse and is time consuming.

In the present example, as in many real-life workplaces, the computer operator must repeat the same operations with respect to the same types of files all day long. Specifically, for each customer signature card, the operator opens three windows and repositions and resizes them from the original default conditions as shown in Figure 1 to the desirable conditions shown in Figure 2.

In accordance with the present invention, the application program is provided with software which "observes" each file that the user opens, records one or more of the display attributes of that file as set by the user and then applies those attributes each time a file of the same type is opened. This feature is herein termed attribute coercion.

A table is maintained in memory in which each of the display attributes, such as window position and window size, are stored in association with the file type. The table may be stored in a RAM memory or a persistent memory device such

as a tape or a disk. The file type can be indicated in many forms, including, but not limited to, the extension or extensions of the file name, such as .gif and .back.gif. In one preferred embodiment, a table is maintained in memory in which an entry for each particular file type is updated every time an operator closes a file of that given type. Particularly, when an operator closes a file of a particular type, the display attributes applied to that file at the time it is closed are stored to the table.

The operator should have the option of disabling the attribute coercion feature on a per-instance basis as well as generally. In addition, in a preferred embodiment, the feature can be selectively disabled with respect to particular file types.

Even further, while it is envisioned that the invention is incorporated as part of the application program or programs with which it will be used, it also can be provided as a separate software package that can operate generally with some or all of the other application programs loaded on the same computer.

Below is an exemplary algorithm in accordance with the present invention.

- The inventive software program creates a master attributes file when it is installed.
- Whenever a file is closed, an entry is placed in the master attributes file. The entry would include the

extension or extensions of the file and the attributes of interest. In this example, the attribute of interest will be screen size. However, it should be understood that any attribute and any number of attributes can be stored associated with the file type. For example, after closing acme.aoi.gif, the following entry is added to the table:

```
.aoi.gif      size      850x525
```

Where 850x525 is the window size of aoi.gif at the time it was closed. This entry indicates that all files bearing the extensions .aoi.gif should be opened in a window of size 850 pixels x 525 pixels.

- Similarly, after closing smith.sig.gif, the following entry is added to the master attributes file:

```
.sig.gif      size      50x75
```

- When a file is opened, the program searches for an entry in the table for the same extension or extensions of the file that is being opened. If one is located, the file is opened at the size indicated in the table.
- The operator may disable attribute coercion for particular document types. For example, the user may be provided with a menu option for disabling attribute coercion by file type. For instance, if the operator disables attribute coercion for .sig.gif files, the following entry is added in the master attributes table:

```
.sig.gif      *
```

This designation is never overwritten by the application unless the user removes the restriction.

- Additional attributes may be associated with each document type. For example, if a document type should be characterized by multiple views, the following entry may be placed in the table:

.sig.gif	size	50x75	
.sig.gif	related	#N.back.gif	size 50x75

This implies that, whenever a file of type .sig.gif is opened, it should be placed into a window of size 50x75. Moreover, the second line indicates that whenever a file bearing the .sig.gif extension is opened, a second file bearing the same name, but having the extension .back.gif should be opened with size 50x75. Thus, when the user indicates to open smith.sig.gif, the software opens both smith.sig.gif and smith.back.gif.

Other attributes besides window size which may be most useful to store in the master attributes table include window position, window color, editing mode, and default font.

In accordance with another aspect of the invention, rather than relying on stored information about the file type, display attributes to be applied to a file can be automatically selected based on data available within the file itself. For instance, with respect to HTML documents, it may be desirable to open the documents with a size that is

customized based on the number of lines in the HTML file.

Thus, for example, software in accordance with the invention may count the number of lines in a file during the process of opening the file and then automatically size the window in which it opens to the minimum size which allows all of the lines to be displayed at once. If the number of lines exceeds the full size of the screen, the window is simply opened at its largest possible size. Preferably, it also displays the number of lines which are not in the viewing area when the number of lines is greater than can be shown in the available screen area.

This feature also may be extended to drop down boxes, including drop down menus. Specifically, a drop down box typically displays x number of elements, when x is an integer. So if the list in the drop down box contains $x+1$ elements, the user is forced to scroll to see that last element. However, the invention can be adapted to determine the number of lines in the box and size the drop down box to a size which will display all of the elements in that box. This feature can be applied to drop down menus in application programs and to drop down boxes in Web pages.

Even further, the invention can be used in connection with a list box, such as an HTML list box, that displays a list of elements.

The invention provides for easy manipulation of a program interface. A software package in accordance with the present

invention may provide a properties sheet window in which the operator can customize operation of the invention by setting such parameters as (1) file types for which the auto attribute coercion is disabled, (2) file types for which the assigned
5 attributes are manually set by the operator and not further modified except by the operator manually and the set of attributes that will and will not be coerced, (3) the set of attributes that will and will not be assigned by file type, and (4) when the attributes will be assigned (eg, when the
10 file is closed automatically or when the operator manually chooses, such as by a clicking a tool bar button).

The invention would be particularly useful in connection with programs that are typically used with a limited number of file types and for which the users of the program commonly
15 perform repetitive tasks with different files of the same type.

In the examples used herein, the file type is based on the file name extension or extensions. Thus, in these examples, anything after the first period in the file name is
20 considered an extension and can be used to designate file type. It should be understood by those of skill in the art that other means for identifying files by type can be implemented in accordance with the invention. For example, rather than file name extension, the program might use
25 attribute information contained within the file itself in a known location. Thus, when the file is being opened, these

attributes can be read from the designated location and used to determine file type.

Figure 3 is a flow chart illustrating operation of the invention in accordance with a first particular embodiment.

5 It should be understood that this flow chart represents merely an exemplary embodiment and that many modifications are possible, including some of those modifications discussed above.

10 The process begins when a file is opened in step 300. In step 310, the program determines the file type, such as by observing the extension or extensions in the file name. In step 320, it is determined whether there is an entry in the table for this file type. If not, flow proceeds to step 340 where the default attributes of the application program with which the invention is being utilized are applied. However, 15 if there is an entry in the table, flow instead proceeds to step 330 where the attributes assigned to that file type in the table are used for opening this file.

20 In step 350, it is determined whether the operator has closed the file. If not, flow proceeds to step 360 to wait for the file to be closed. It should be understood by those of skilled in the art, that step 350 and 360 are merely exemplary and that, in other embodiments, the file closing event might be indicated by an interrupt rather than by 25 continuously looping through steps 350 and 360.

When the file is eventually closed, in step 370, it is determined whether the attribute coercion feature is enabled. If not, flow proceeds directly to the end at step 390. However, if it is enabled, flow proceeds to step 380 where the table is updated with the values of the specific attributes that exist at that time.

Figure 4 illustrates operation of the invention in accordance with a second embodiment. The process begins at step 400 when a file is opened. In step 410, the file properties of interests, such as the number of lines in the file, the number of lines in one or more drop boxes in the file, and/or the number of lines in one or more list boxes in the file are determined. This can be accomplished by examining the source code for the file.

In step 420, the relevant attributes, such as window size, drop box size and/or list box size are customized based on the pertinent line count. The window size in which a file will be opened is dictated by the application program which is being used to open the file. Therefore, the inventive method and apparatus is applied within the application program with which it is being used without the need to modify the file that is being opened. However, with respect to display attributes that are normally dictated by the contents of the file itself such the size of a drop box or list box within an HTML document, the actual file must be modified in order to effect the present invention. In at least one preferred

embodiment, the customization step comprises revising the source code within the file as it is stored in RAM for execution. OF course, preferably, no change is made to the original copy of the file. The process ends in step 430.

5 Having thus described a few particular embodiments of the invention, various alterations, modifications, and improvements will readily occur to those skilled in the art. Such alterations, modifications and improvements as are made obvious by this disclosure are intended to be part of this description though not expressly stated herein, and are intended to be within the spirit and scope of the invention. Accordingly, the foregoing description is by way of example only, and not limiting. The invention is limited only as defined in the following claims and equivalents thereto.

10

15

CLAIMS

I claim:

1. A method of providing an interface with displayable computer files on a computer display, said method comprising
5 the steps of:

(1) displaying a first displayable file on said computer display in a manner customized by an operator of said computer;

(2) storing data indicating a value of at least one attribute of the manner in which said first file was displayed associated with data indicating a type of said first file; and
10

(3) when a next file of the type of said first file is opened by an operator for display, displaying said next file having the same value of said attribute as said first file.
15

2. The method of claim 1 wherein step (2) comprises storing said value when said first file is closed by said operator.

3. The method of claim 2 wherein said value that is stored in step (2) is the value of said attribute when said first file was closed.
20

4. The method of claim 1 wherein the value that is stored in step (2) is the value of said attribute at a time selected by said operator.

5 5. The method of claim 1 wherein said attribute is a size of a window within which said first file was displayed.

6. The method of claim 5 wherein said attribute further comprises a position of said window within said computer display.

10 7. The method of claim 1 wherein said attribute further comprises a position of said window within said computer display.

8. The method of claim 1 wherein said file type comprises a file name extension of said first file.

15 9. The method of claim 8 wherein said extension comprises multiple extensions.

10. The method of claim 8 wherein step (3) comprises reading said table to determine said attribute.

11. The method of claim 1 wherein said storing step comprises storing said data as an attribute table comprising a

plurality of data entries relating file types to attribute values.

12. The method of claim 1 further comprising the steps of:

5 (4) displaying a second displayable file simultaneously with said first file on said computer display in a manner selected by said operator;

10 (5) storing data associated with said type of said first file indicating at least a type of said second file relative to said first file; and

15 (6) when a next file of the type of said first file is opened for display, automatically opening a next file of the same type as said second file and having the same relationship to said next file as said second file had to said first file.

13. The method of claim 12 wherein step (5) comprises storing said type of said second file when said first file is closed by said operator.

20 14. The method of claim 13 further comprising the steps of:

(7) storing data indicating a value of at least one attribute of said second file, said data associated with said

type of said second file and with said type of said first file; and

(8) when a next file of the same type as said first file is opened by an operator, assigning said next file of the same type as said second file the same value for said attribute of said second file.

15. The method of claim 14 wherein step (7) comprises storing said value when said first file is closed by said operator.

16. The method of claim 15 wherein said value that is stored in step (7) is the value of said attribute when said first file was closed.

17. The method of claim 12 wherein the value that is stored in step (7) is the value of said attribute at a time selected by said operator.

18. The method of claim 14 wherein said attribute of said second file comprises a size of a window within which said second file was displayed.

19. The method of claim 18 wherein said attribute of said second file further comprises a position of said window.

20. The method of claim 14 wherein said attribute of said second file further comprises a position of said window.

5 21. A computer program for providing an interface with displayable computer files on a computer display, said program comprising:

(1) means for determining a value of at least one display attribute of a first displayable file on a computer;

(2) means for determining a type of said first file;

10 (3) means for storing said value associated with data indicating said type of said first file; and

(3) means for displaying a next file of the same type as said first file using the same value of said attribute as said first file when a next file of the type of said first file is opened.

15 22. The computer program of claim 21 wherein said means for storing comprises means for storing said value of said attribute when said first file was closed.

23. The computer program of claim 22 wherein said file type comprises a file name extension of said first file.

20 24. The computer program of claim 21 wherein said means for storing comprises means for storing said data as an

attribute table comprising a plurality of data entries relating file types to attribute values.

25. The computer program of claim 21 further comprising:
means for determining a value of at least one
5 display attribute of a second displayable file on a computer
that is displayed simultaneously with said first file;
means for determining a type of said second file;
means for storing data associated with said type of
said first file indicating at least said type of said second
10 file relative to said first file and said at least one
attribute of said second file; and
means for automatically opening a next file of the
same type as said second file and having the same relationship
to said next file as said second file had to said first file
15 when a next file of the type of said first file is opened for
display.

METHOD AND APPARATUS FOR LEARNING
COMPUTER INTERFACE ATTRIBUTES

Abstract of the Invention

5 The invention is a method and apparatus for automatically
opening files of particular types in a computer application
program using certain attributes, such as window size and
position, as dictated by user preferences based on the user's
previous habits in opening files of the same type in that
particular computer application program or based on properties
10 of the file itself.

M:\TNaccarella\CLIENTS\IBM\23,845\Pat Office\Application23845.wpd

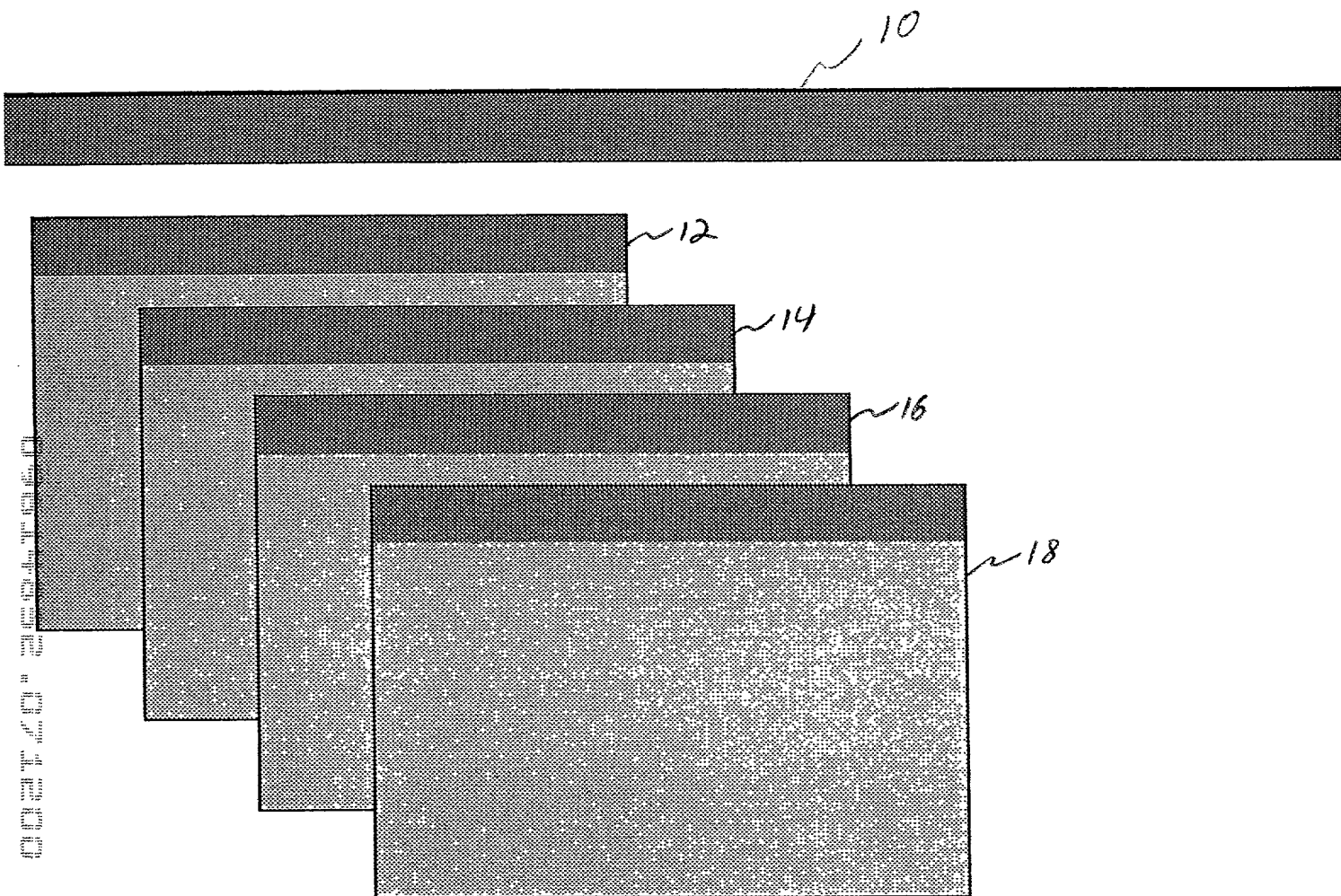


Figure 1

N10

09614852 071200

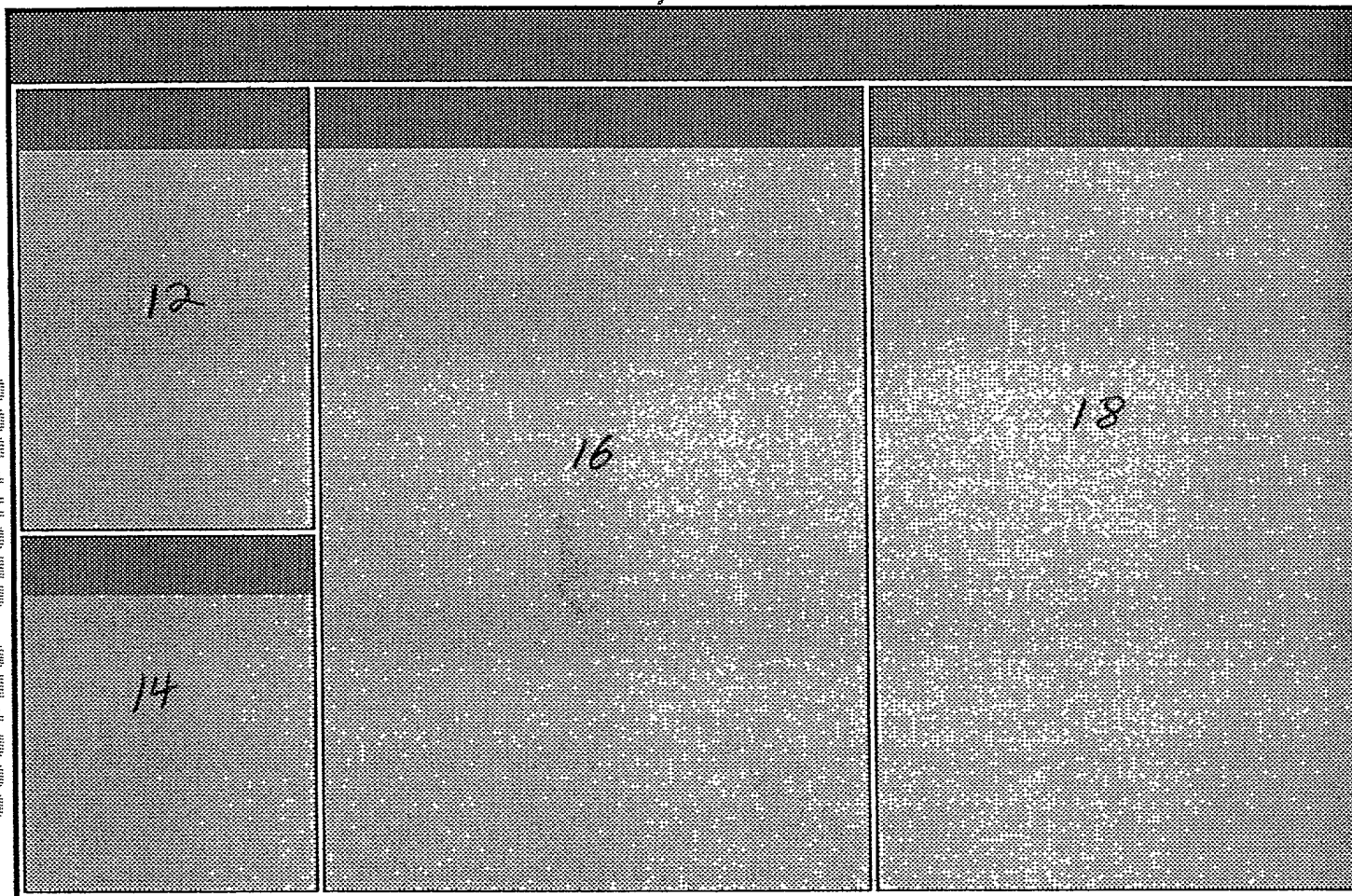


Figure 2

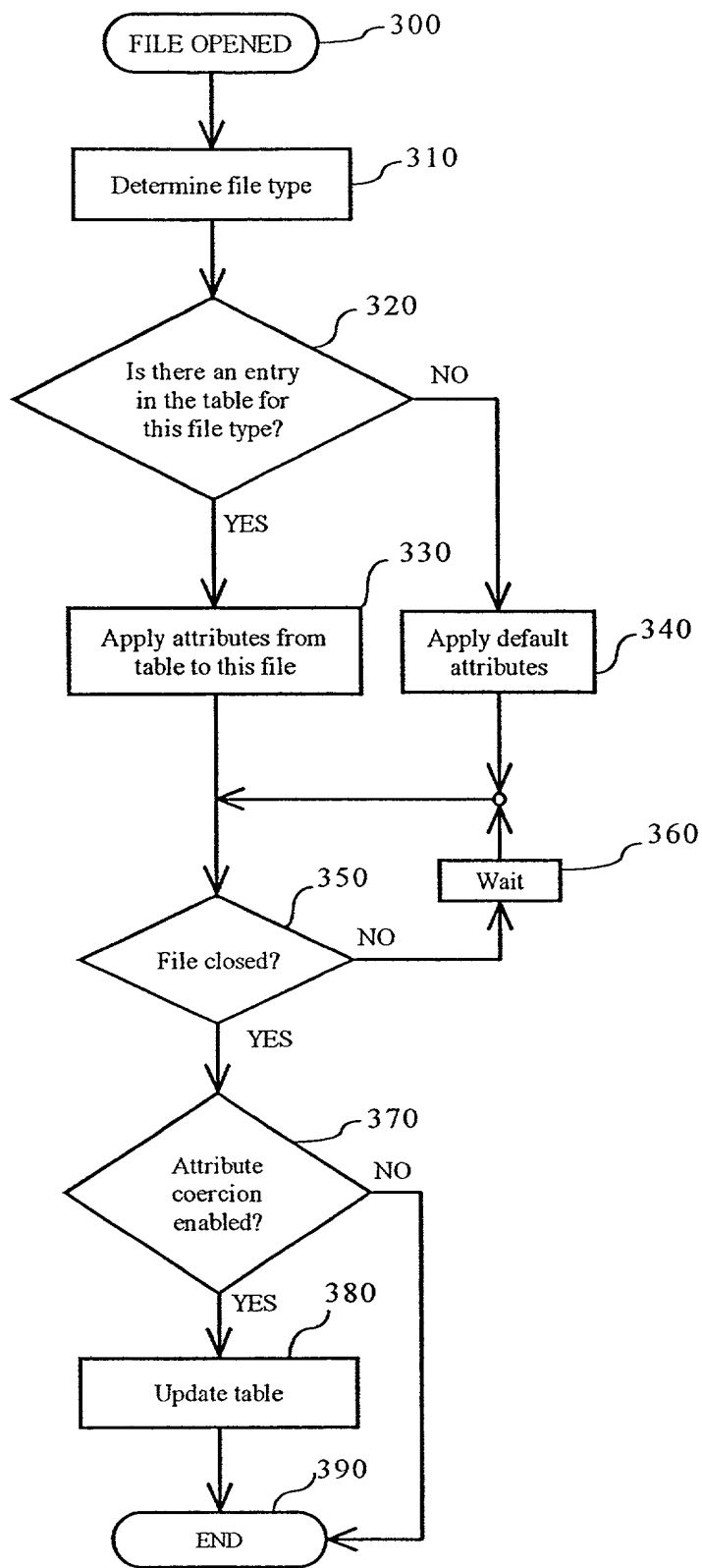


Fig. 3

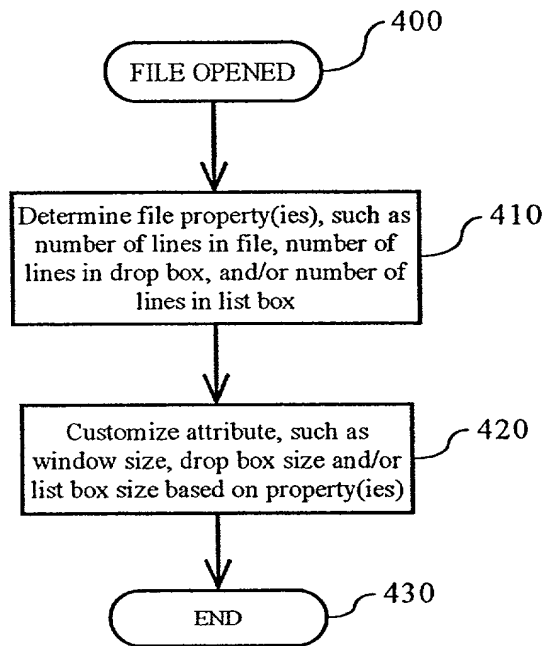


Fig. 4

Declaration and Power of Attorney for Patent Application

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

METHOD AND APPARATUS FOR LEARNING COMPUTER INTERFACE ATTRIBUTES

the specification of which:

☒ is attached hereto.

☐ was filed on _____ as Application Serial No. _____ and was amended on _____.

I hereby state that I have reviewed and understand the contents of the above- identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):			
Number	Country	Day/Month/Year	Priority Claimed
_____	_____	_____	_____

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Prior U.S. Applications:
Serial No.

Filing Date

Status

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

A.B. Clay	Reg. No. 32,121
G. M. Doudnikoff	Reg. No. 32,847
E. H. Duffield	Reg. No. 25,970
J. W. Herndon	Reg. No. 27,901
J. S. Ray-Yarletts	Reg. No. 39,808.
John T. Synnestvedt	Reg. No. 18,117
Charles H. Lindrooth	Reg. No. 20,659
Irving Newman	Reg. No. 22,638
Alexis Barron	Reg. No. 22,702
Peter J. Butch, III	Reg. No. 32,203
Joseph F. Posillico	Reg. No. 32,290
Mark D. Simpson	Reg. No. 32,942
Theodore Naccarella	Reg. No. 33,023
Patrick J. Kelly, Ph.D.	Reg. No. 34,638
Gary A. Hecht	Reg. No. 36,826
Stephen J. Driscoll	Reg. No. 37,564
Lisa B. Lane	Reg. No. 38,217
Joshua R. Slavitt	Reg. No. 40,816
John A. Chionchio	Reg. No. 40,954
Gregory S. Bernabeo	Reg. No. 44,032
Stephen J. Weed	Reg. No. 45,202

Send all correspondence to:

Theodore Naccarella, Esquire
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107-2950

Telephone: (215) 923-4466
Facsimile: (215) 923-2189

Inventor: **Karen R. Kluttz**

Signature:  Date: 6/16/00

Residence: **108 West Aycock Street, Raleigh, NC 27608**

Citizenship: **United States**

Post Office Address: Same as residence

Inventor: **Sandeep K. Singhal**

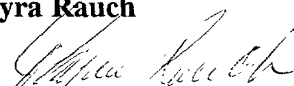
Signature:  Date: 16 June 2000

Residence: **1021-105 Brighthurst Drive, Raleigh, NC 27605**

Citizenship: **United States**

Post Office Address: Same as residence

Inventor: **Thyra Rauch**

Signature:  Date: July 5, 2000

Residence: **634 North Blount Street, Raleigh, NC 27604**

Citizenship: **United States**

Post Office Address: Same as residence

Inventor: **David A. Schell**

Signature:  Date: July 10, 2000

Residence: **1103 Hamilton Way, Durham, NC 27713**

Citizenship: **United States**

Post Office Address: Same as residence

M:\TNaccarella\CLIENTS\IBM\23,845\Pat Office\Declaration.wpd